

---

# **pynwn Documentation**

***Release 0.1***

**jmd**

March 10, 2016



<b>1</b>	<b>Objects</b>	<b>3</b>
1.1	Area . . . . .	3
1.2	Creature . . . . .	6
1.3	Dialog . . . . .	9
1.4	Door . . . . .	10
1.5	Encounter . . . . .	13
1.6	Faction . . . . .	14
1.7	Item . . . . .	14
1.8	Journal . . . . .	16
1.9	Module . . . . .	17
1.10	Palette . . . . .	18
1.11	Placeable . . . . .	18
1.12	Player Character . . . . .	21
1.13	Sound . . . . .	24
1.14	Store . . . . .	26
1.15	Trigger . . . . .	27
1.16	Waypoint . . . . .	29
<b>2</b>	<b>Resources</b>	<b>31</b>
2.1	Container . . . . .	31
2.2	ContentObject . . . . .	32
2.3	DirectoryContainer . . . . .	32
2.4	ResourceManager . . . . .	33
<b>3</b>	<b>File Formats</b>	<b>35</b>
3.1	2da . . . . .	35
3.2	2dx . . . . .	35
3.3	Erf . . . . .	36
3.4	Gff . . . . .	37
3.5	Key / Bif . . . . .	38
3.6	Tlk . . . . .	38
3.7	Tls . . . . .	39
<b>4</b>	<b>Checking Local Variables</b>	<b>41</b>
<b>5</b>	<b>Dialog Printer</b>	<b>43</b>
<b>6</b>	<b>Minimap Generator</b>	<b>45</b>

<b>7 Indices and tables</b>	<b>47</b>
<b>Python Module Index</b>	<b>49</b>

Contents:



## 1.1 Area

`class pynwn.Area(resref, container)`

**`__weakref__`**

list of weak references to the object (if defined)

**`comment`**

Comments.

**`creatures`**

Creature instances.

**Returns** List of `pynwn.CreatureInstance` objects.

**`day_ambient_color`**

Day ambient color.

**`day_diffuse_color`**

Day diffuse color.

**`day_fog_amount`**

Daytime fog amount (0-15)

**`day_fog_color`**

Daytime fog color (BGR format)

**`day_night_cycle`**

1 if day/night transitions occur, 0 otherwise.

**`day_shadows`**

1 if shadows appear during the day, 0 otherwise.

**`doors`**

Door instances.

**Returns** List of `DoorInstance` objects.

**`encounters`**

Encounters

**Returns** List of `pynwn.EncounterInstance` objects.

**`flags`**

Bit flags specifying area terrain type.

**fog\_clip\_distance**

Fog clip distance.

**get\_name** (*lang=None*)

Localized name.

**height**

Area height.

**is\_night**

1 if always night, 0 if always day.

**items**

Item instance list.

**Returns** List of *pynwn.ItemInstance* objects.

**lighting\_scheme**

Index into environment.2da

**lightning**

Chance of lightning [0,100]

**listen\_modifier**

Modifier to Listen skill checks made in area

**load\_screen**

Load screen ID

**night\_ambient\_color**

Nighttime ambient color.

**night\_diffuse\_color**

Nighttime diffuse color.

**night\_fog\_amount**

Nighttime fog amount (0-15)

**night\_fog\_color**

Nighttime fog color.

**night\_shadows**

1 if shadows appear at night, 0 otherwise

**no\_rest**

1 if resting is not allowed, 0 otherwise

**placeables**

Placeables

**Returns** List of *pynwn.PlaceableInstance* objects.

**pvp**

Area PvP setting.

**rain**

Chance of rain [0,100]

**resref**

Resref.

**scripts**

Scripts. Responds to script events:

1.Event.ENTER



2.Event.EXIT

3.Event.HEARTBEAT

4.Event.USER\_DEFINED

**set\_name** (*lang=None, string=None*)

Localized name.

**shadow\_opacity**

Opacity of shadows (0-100).

**skybox**

Index into skyboxes.2da.

**snow**

Chance of snow [0,100]

**sounds**

Sounds

**Returns** List of *pynwn.SoundInstance* objects.

**spot\_modifier**

Modifier to Spot skill checks made in area

**stores**

Stores

**Returns** List of *pynwn.StoreInstance* objects.

**tag**

Tag.

**tiles**

Tiles :returns: List of *pynwn.TileInstance* objects.

**tileset**

Tileset.

**triggers**

Triggers

**Returns** List of *pynwn.TriggerInstance* objects.

**vars**

Variable table

**version**

Area version

**waypoints**

Waypoints

**Returns** List of *pynwn.WaypointInstance* objects.

**width**

Area width.

**wind\_power**

Strength of the wind in the area. None, weak, or strong (0-2).

## 1.2 Creature

**class** `pynwn.Creature` (*resource*, *instance=False*)

The Creature class abstracts over UTCs only. It doesn't handle all the additional fields one finds in BICs, see `pynwn.PlayerCharacter` for that.

**Parameters**

- **resource** (str, (`pynwn.ContentObject`, `pynwn.Container`), or `pynwn.GffInstance`) – Filename or content object.
- **instance** – Determines if the current creature is an instanced object or not.

**appearance**

Appearance ID.

**bodybag**

Body bag.

**charisma**

Creature's charisma.

**classes**

Creature's classes.

**Returns** List of tuples containing class ID and level.

**comment**

Comment.

**constitution**

Creature's constitution.

**conversation**

Dialog resref.

**corpse\_decay**

Corpse decay time.

**cr**

Challenge Rating

**dexterity**

Creature's dexterity.

**diety**

Deity

**disarmable**

Disarmable flag.

**equips**

Creature's equipment list.

**Returns** List of tuples containing equipment ID and resref.

**faction**

Faction ID.

**feats**

Creature's feats.

**Returns** List of feat IDs.

**gender**  
Gender.

**get\_description** (*lang=None*)  
Localized description

**get\_name\_first** (*lang=None*)  
Localized first name

**get\_name\_last** (*lang=None*)  
Localized last name

**get\_skill** (*skill*)

**goodevil**  
Good - Evil

**hp**  
HP.

**hp\_current**  
Current hitpoints.

**hp\_max**  
Maximum hitpoints.

**intelligence**  
Creature's intelligence.

**interruptable**  
Conversation interruptable flag.

**is\_immortal**  
Immortal flag.

**is\_pc**  
Player character flag.

**items**  
Creature's inventory items.

**Returns** List of RepositoryItems.

**lawchaos**  
Lawful - Chaotic

**lootable**  
Lootable.

**natural\_ac**  
Natural AC.

**no\_perm\_death**  
No permanent death flag.

**palette\_id**  
Palette ID.

**perception\_range**  
Perception Range.

**phenotype**  
Phenotype ID.

**portrait\_id**  
Portrait ID.

**race**  
Racial Type

**resref**  
Resref.

**save\_fortitude**  
Fortitude saving throw.

**save\_reflex**  
Reflex saving throw.

**save\_will**  
Will saving throw.

**scripts**  
Creature's scripts. Responds to script events:

- Event.HEARTBEAT
- Event.PERCEPTION
- Event.SPELL\_CAST\_AT
- Event.ATTACKED
- Event.DAMAGED
- Event.DISTURBED
- Event.END\_COMBAT\_ROUND
- Event.CONVERSATION
- Event.SPAWN
- Event.REST
- Event.DEATH
- Event.USER\_DEFINED
- Event.BLOCKED

**set\_description** (*lang=None, string=None*)  
Localized description

**set\_name\_first** (*lang=None, string=None*)  
Localized first name

**set\_name\_last** (*lang=None, string=None*)  
Localized last name

**set\_skill** (*skill, value*)

**skills**  
Creature's skills

**Returns** List of skill ranks in order of skill ID.

**special\_abilities**

**stage** ()  
Stages changes creature's GFF structure.

**starting\_package**  
Starting package ID.

**strength**  
Creature's strength.

**subrace**  
Subrace.

**tag**  
Tag

**tail**  
Tail ID.

**vars**  
Variable table

**walkrate**  
Walkrate.

**wings**  
Wings ID.

**wisdom**  
Creature's wisdom.

---

**class** `pynwn.CreatureInstance` (*gff, parent\_obj*)

A creature instance is one placed in an area in the toolset. As such it's values are derived from its parent GFF structure.

**equips**  
Creature's equipment list.

**Returns** List of tuples containing equipment ID and `pynwn.ItemInstance`.

**items**  
Creature's inventory items.

**Returns** List of tuples contain repository position and the `pynwn.ItemInstance`.

**stage** ()  
Stages changes to the creature instances parent object.

## 1.3 Dialog

**class** `pynwn.Dialog` (*resource*)  
Abstracts .dlg GFF files.

**delay\_entry**  
DelayEntry.

**entries**  
Entries

**Returns** List of DialogNodes

**prevent\_zoom**  
No zoom flag.

**replies**

Replies

**Returns** List of DialogNodes**resref**

Resref

Dialogs don't store their resref internally.

**script\_abort**

Conversation abort script.

**script\_end**

Conversation end script.

**stage()**

Stages changes to the dialog's GFF structure.

**starts**

Starts

These are limited pointers in the entry list to the topmost level of dialog in a concersation.

**Returns** List of DialogPointers**word\_count**

Word count.

## 1.4 Door

**class** `pynwn.Door` (*resource, instance=False*)**animation\_state**

Animation State.

**appearance**

Appearance ID.

**comment**

Comment.

**conversation**

Conversation resref.

**faction**

Faction ID.

**generic\_type**

Generic type.

**get\_description** (*lang=None*)

Localized description.

**get\_name** (*lang=None*)

Localized name.

**hardness**

Hardness.

**hp**  
Maximum Hitpoints.

**hp\_current**  
Current Hitpoints.

**interruptable**  
Conversation interruptable flag.

**key\_auto\_remove**  
Auto-remove key.

**key\_required**  
Key required flag.

**key\_tag**  
Key tag.

**linked\_to**  
Linked to tag.

**linked\_to\_flags**  
Linked to flags.

**load\_screen**  
Load screen ID.

**lock\_close\_dc**  
Close lock DC.

**lock\_open\_dc**  
Open lock DC.

**lockable**  
Lockable flag.

**locked**  
Locked flag.

**paletted\_id**  
Palette ID.

**plot**  
Plot flag.

**portrait\_id**  
Portrait ID.

**resref**  
Resref.

**save\_fortitude**  
Fortitude Saving Throw.

**save\_reflex**  
Reflex Saving Throw.

**save\_will**  
Will Saving Throw.

**scripts**  
Scripts: Door responds to the following script events:  
1.Event.CLOSE

2.Event.DAMAGED  
3.Event.DEATH  
4.Event.TRAP\_DISARMED  
5.Event.HEARTBEAT  
6.Event.LOCK  
7.Event.ATTACKED  
8.Event.OPEN  
9.Event.SPELL\_CAST\_AT  
10.Event.TRAP\_TRIGGERED  
11.Event.UNLOCK  
12.Event.USER\_DEFINED  
13.Event.CLICK  
14.Event.FAIL\_TO\_OPEN

**set\_description** (*lang=None, string=None*)  
Localized description.

**set\_name** (*lang=None, string=None*)  
Localized name.

**tag**  
Tag.

**trap\_detect\_dc**  
Trap detect DC.

**trap\_detectable**  
Trap detectable flag.

**trap\_disarm\_dc**  
Trap disarm DC.

**trap\_disarmable**  
Trap disarmable flag.

**trap\_flag**  
Trap flag.

**trap\_one\_shot**  
Trap is one-shot.

**trap\_type**  
Trap type.

**vars**  
Variable table

---

**class** pynwn.**DoorInstance** (*gff, parent\_obj*)

A door instance is one placed in an area in the toolset. As such it's values are derived from its parent GFF structure.

**position**  
Position



**Returns** Tuple of x, y, z coordinates.

## 1.5 Encounter

`class pynwn.EncounterCreature (gff, parent_obj)`

**appearance**  
Appearance

**cr**  
Challenge rating

**resref**  
Resref

**unique**  
Unique spawn.

—

`class pynwn.Encounter (resource, instance=False)`

**active**  
Active flag.

**comment**  
Comment.

**creatures**  
Creatures in the encounter.

**Returns** List of *EncounterCreature* objects.

**difficulty**  
Difficulty.

**difficulty\_index**  
Difficulty Index.

**faction**  
Faction ID.

**get\_name** (*lang=None*)  
Localized name.

**max\_creatures**  
Maximum creatures.

**palette\_id**  
Palette ID.

**player\_only**  
Triggered by player only.

**rec\_creatures**  
rec\_creatures.

**reset**  
Resets flag.

**reset\_time**  
Reset time.

**respawns**  
Respawns.

**resref**  
Resref.

**scripts**  
Scripts. Responst to script events:

- Event.ENTER
- Event.EXIT
- Event.EXHAUSTED
- Event.HEARTBEAT
- Event.USER\_DEFINED

**set\_name** (*lang=None, string=None*)  
Localized name.

**spawn\_option**  
Spawn option.

**tag**  
Tag.

**vars**  
Variable table

---

**class** `pynwn.EncounterInstance` (*gff, parent\_obj*)  
A encounter instance is one placed in an area in the toolset. As such it's values are derived from its parent GFF structure. It never needs to be instantiated directly.

## 1.6 Faction

**class** `pynwn.Faction` (*resource*)

**factions**

**reputations**

**stage** ()  
Stages changes to GFF structure.

## 1.7 Item

**class** `pynwn.Item` (*resource, instance=False*)

**base\_type**  
Base item ID.

**charges**  
Charges.

**comment**  
Comment.

**cost**  
Cost.

**cost\_additional**  
Additional Cost.

**cursed**  
Cursed flag.

**display\_name**  
Display Name.

**get\_description** (*lang=None*)  
Localized unidentified description.

**get\_description\_id** (*lang=None*)  
Localized identified description.

**get\_name** (*lang=None*)  
Localized name.

**identified**  
Identified flag.

**palette\_id**  
Palette ID.

**plot**  
Plot flag.

**properties**  
Item properties  
  
Returns List of *ItemProperty* objects.

**resref**  
Resref.

**set\_description** (*lang=None, string=None*)  
Localized unidentified description.

**set\_description\_id** (*lang=None, string=None*)  
Localized identified description.

**set\_name** (*lang=None, string=None*)  
Localized name.

**stack\_size**  
Stack size.

**stolen**  
Stolen flag.

**tag**  
Tag.

**vars**  
Variable table

---

```
class pynwn.ItemInstance (gff, parent_obj)
```

A item instance is one placed in an area in the toolset. As such it's values are derived from its parent GFF structure. It never needs to be instantiated directly.

---

```
class pynwn.ItemProperty (gff, parent_obj)
```

**chance**

Appearance chance.

**cost\_table**

Cost table.

**cost\_value**

Cost value.

**param\_table**

Parameter table.

**param\_value**

Parameter value.

**subtype**

Subtype.

**type**

Type.

---

```
class pynwn.RepositoryItem (gff, parent_obj)
```

**dropable**

Dropable flag.

**infinite**

Infinite flag.

**position**

Position in inventory

**Returns** Tuple of x and y coordinates.

**resref**

Resref.

## 1.8 Journal

```
class pynwn.Journal (resource)
```

There is only one journal file.

**quests**

## 1.9 Module

**class** `pynwn.Module` (*module*)

Module abstracts over MOD ERF files and directories containing the contents of MOD files.

**areas**

Areas.

**Returns** List of `pynwn.Area` objects.

**entry\_area**

Entry area.

**Returns** `pynwn.Area` instance.

**entry\_location**

Entry location.

**Returns** Tuple of the X, Y, Z coordinates.

**expansion\_pack**

Expansion pack.

**game\_version**

Game version

**get\_description** (*lang=None*)

Localized description.

**get\_name** (*lang=None*)

Localized name.

**glob** (*glob\_pattern*)

Returns a list of (ContentObject, Container) tuples for file names matching the glob pattern. i.e. Unix shell-style wildcards: \*.utc Note: all file names are converted to lowercase.

**haks**

List of HAK files without 'hak' extension.

**scripts**

Scripts. Responds to script events:

1.Event.CUTSCENE\_ABORT

2.Event.ENTER

3.Event.EXIT

4.Event.HEARTBEAT

5.Event.ITEM\_ACQUIRED

6.Event.ITEM\_ACTIVATED

7.Event.ITEM\_EQUIPPED

8.Event.ITEM\_UNACQUIRED

9.Event.ITEM\_UNEQUIPPED

10.Event.LEVELUP

11.Event.LOAD

12.Event.DEATH

13.Event.DYING  
14.Event.RESPAWN  
15.Event.REST  
16.Event.USER\_DEFINED  
**set\_description** (*lang=None, string=None*)  
Localized description.  
**set\_name** (*lang=None, string=None*)  
Localized name.  
**stage** ()  
**tlk**  
Custom TLK file without file extension.  
**vars**  
Variable table  
**xp\_scale**  
Experience point scale.

## 1.10 Palette

**class** pynwn.**PaletteNode** (*gff, parent\_obj*)  
Palette Node  
  
**cr**  
**faction**  
**id**  
**is\_leaf**  
**name**  
**nodes**  
**resref**  
**stage** ()  
**strref**  
  
**class** pynwn.**Palette** (*resource*)  
This is a very rough abstraction over ITPs.  
  
**nodes**  
Gets all nodes in the palette.  
  
    **Returns** a list of *PaletteNode* objects.  
  
**stage** ()

## 1.11 Placeable

**class** pynwn.**Placeable** (*resource, instance=False*)

**animation\_state**  
Animation State.

**appearance**  
Appearance ID.

**body\_bag**  
Body bag.

**comment**  
Comment.

**conversation**  
Conversation resref.

**faction**  
Faction ID.

**get\_description** (*lang=None*)  
Localized description.

**get\_name** (*lang=None*)  
Localized name.

**hardness**  
Hardness.

**has\_inventory**  
Has inventory flag.

**hp**  
Maximum Hitpoints.

**hp\_current**  
Current Hitpoints.

**interruptable**  
Conversation interruptable flag.

**items**  
Inventory items.

**Returns** List of RepositoryItem objects or [] if the object does not have an inventory.

**key\_auto\_remove**  
Auto-remove key.

**key\_required**  
Key required flag.

**key\_tag**  
Key tag.

**lock\_close\_dc**  
Close lock DC.

**lock\_open\_dc**  
Open lock DC.

**lockable**  
Lockable flag.

**locked**  
Locked flag.

**paletted\_id**

Palette ID.

**plot**

Plot flag.

**portrait\_id**

Portrait ID.

**resref**

Resref.

**save\_fortitude**

Fortitude Saving Throw.

**save\_reflex**

Reflex Saving Throw.

**save\_will**

Will Saving Throw.

**scripts**

Scripts. Responds to script events:

- 1.Event.CLOSE
- 2.Event.DAMAGED
- 3.Event.DEATH
- 4.Event.TRAP\_DISARMED
- 5.Event.HEARTBEAT
- 6.Event.LOCK
- 7.Event.ATTACKED
- 8.Event.OPEN
- 9.Event.SPELL\_CAST\_AT
- 10.Event.TRAP\_TRIGGERED
- 11.Event.UNLOCK
- 12.Event.USER\_DEFINED
- 13.Event.CLICK
- 14.Event.DISTURBED
- 15.Event.USED

**set\_description** (*lang=None, string=None*)

Localized description.

**set\_name** (*lang=None, string=None*)

Localized name.

**stage** ()

Stage changes to the placeable's GFF structure.

**static**

Static flag.



**tag**  
Tag.

**trap\_detect\_dc**  
Trap detect DC.

**trap\_detectable**  
Trap detectable flag.

**trap\_disarm\_dc**  
Trap disarm DC.

**trap\_disarmable**  
Trap disarmable flag.

**trap\_flag**  
Trap flag.

**trap\_one\_shot**  
Trap is one-shot.

**trap\_type**  
Trap type.

**type**  
Type.

**useable**  
Useable flag.

**vars**  
Variable table

---

**class** `pynwn.PlaceableInstance` (*gff, original*)

A placeable instance is one placed in an area in the toolset. As such it's values are derived from its parent GFF structure.

**items**  
Inventory items.

**Returns** List of Tupels contain repository position and the ItemInstance.

**position**

**stage** ()  
Stage changes to the placeable instance's parent GFF structure.

## 1.12 Player Character

**class** `pynwn.PlayerCharacter` (*resref, container*)

**appearance**  
Appearance ID.

**appearance\_head**  
Head appearance ID.

**bodybag**  
Body bag.

**charisma**

Creature's charisma.

**constitution**

Creature's constitution.

**conversation**

Dialog resref.

**corpse\_decay**

Corpse decay time.

**cr**

Challenge Rating

**dead\_selectable**

Selectable when dead flag.

**desc\_override**

Description Override

**detect\_mode**

Detect mode.

**dexterity**

Creature's dexterity.

**diety**

Deity

**disarmable**

Disarmable flag.

**equips**

Creature's equipment list.

**Returns** List of tuples containing equipment ID and ItemInstance.

**experience**

Experience.

**faction**

Faction ID.

**gender**

Gender.

**get\_description** (*lang=None*)

Localized description

**get\_name\_first** (*lang=None*)

Localized first name

**get\_name\_last** (*lang=None*)

Localized last name

**goodevil**

Good - Evil

**hp**

HP.

**hp\_current**

Current hitpoints.

**hp\_max**  
Maximum hitpoints.

**hp\_pregame\_current**  
Pregame current hitpoints.

**intelligence**  
Creature's intelligence.

**is\_commandable**  
Commandable flag.

**is\_destroyable**  
Destroyable flag.

**is\_dm**  
Dungeon Master flag.

**is\_immortal**  
Immortal flag.

**is\_pc**  
Player character flag.

**is\_raisable**  
Raisable flag.

**lawchaos**  
Lawful - Chaotic

**level\_stats**  
Player's level stat list.

**lootable**  
Lootable.

**master**  
Master object ID.

**movement\_rate**  
Movement rate.

**natural\_ac**  
Natural AC.

**no\_perm\_death**  
No permanent death flag.

**phenotype**  
Phenotype ID.

**plot**  
Plot flag.

**portrait**  
Portrait resref.

**portrait\_id**  
Portrait ID.

**race**  
Racial Type

**save\_fortitude**  
Fortitude saving throw.

**save\_reflex**  
Reflex saving throw.

**save\_will**  
Will saving throw.

**set\_description** (*lang=None, string=None*)  
Localized description

**set\_name\_first** (*lang=None, string=None*)  
Localized first name

**set\_name\_last** (*lang=None, string=None*)  
Localized last name

**size**  
Size.

**skillpoints**  
Skill points.

**stage** ()  
Stage changes to the placeable's GFF structure.

**starting\_package**  
Starting package ID.

**stealth\_mode**  
Stealth mode.

**strength**  
Creature's strength.

**subrace**  
Subrace.

**tag**  
Tag

**tail**  
Tail ID.

**wings**  
Wings ID.

**wisdom**  
Creature's wisdom.

## 1.13 Sound

**class** `pynwn.Sound` (*resource, instance=False*)

**active**  
Active flag.

**comment**  
Comment.

**continous**  
Continuous flag.

**distance\_max**  
Maximum distance.

**distance\_min**  
Minimum distance.

**elevation**  
Elevation.

**get\_name** (*lang=None*)  
Localized name.

**hours**  
Hours.

**interval**  
Interval.

**interval\_variation**  
Interval variation.

**looping**  
Looping flag.

**palette\_id**  
Palette ID.

**pitch\_variation**  
Pitch variation.

**positional**  
Positional.

**priority**  
Priority.

**random**  
Random.

**random\_position**  
Random position.

**random\_range**

**resref**  
Resref.

**set\_name** (*lang=None, string=None*)  
Localized name.

**sounds**

**stage** ()

**tag**  
Tag.

**times**  
Times.

**volume**  
Volume.

**volume\_variation**

Volume variation.

---

**class** pynwn.**SoundInstance** (*gff, parent\_obj*)

A sound instance is one placed in an area in the toolset. As such it's values are derived from its parent GFF structure.

**stage** ()

Stages changes to parent GFF structure.

## 1.14 Store

**class** pynwn.**Store** (*resource, instance=False*)

**black\_market**

Black market flag.

**comment**

Comment.

**get\_description** (*lang=None*)

Localized unidentified description.

**get\_description\_id** (*lang=None*)

Localized identified description.

**get\_name** (*lang=None*)

Localized name.

**gold**

Gold.

**items**

Items in inventory.

**Returns** a two dimensional array with the format: [<store page>][<RepositoryItem objects>]

**mark\_down**

Mark down.

**mark\_down\_bm**

Blackmarket mark down.

**mark\_up**

Mark up.

**palette\_id**

Resref.

**price\_id**

Price to identify item.

**price\_max\_buy**

Maximum buy price.

**resref**

Resref.

**scripts**

Scripts. Responds to script events:

1.Event.OPEN

2.Event.CLOSE

**set\_description** (*lang=None, string=None*)

Localized unidentified description.

**set\_description\_id** (*lang=None, string=None*)

Localized identified description.

**set\_name** (*lang=None, string=None*)

Localized name.

**stage** ()

**tag**

Tag.

**vars**

Variable table

**will\_not\_buy**

Will not buy list.

**Returns** List of baseitem IDs that store will not buy.

**will\_only\_buy**

Will only buy list.

**Returns** List of baseitem IDs that store will only buy.

---

**class** pynwn.**StoreInstance** (*gff, parent\_obj*)

A store instance is one placed in an area in the toolset. As such it's values are derived from its parent GFF structure.

**items**

Items in inventory.

**Returns** a two dimensional array with the format: [<store page>][<ItemInstance objects>]

**stage** ()

Stages changes to parent GFF structure.

---

## 1.15 Trigger

**class** pynwn.**Trigger** (*resource, instance=False*)

**comment**

Comment.

**faction**

Faction ID.

**get\_name** (*lang=None*)

Localized name.

**highlight\_height**  
Cursor.

**key\_auto\_remove**  
Auto-remove key flag.

**key\_tag**  
Key tag.

**linked\_to**  
Linked to object tag.

**linked\_to\_flags**  
Linked to flags.

**load\_screen**  
Load screen ID.

**paletted\_id**  
Palette ID.

**portrait\_id**  
Portrait ID.

**resref**  
Resref.

**scripts**  
Scripts. Responds to script events:

- 1.Event.TRAP\_DISARMED
- 2.Event.TRAP\_TRIGGERED
- 3.Event.CLICK
- 4.Event.HEARTBEAT
- 5.Event.ENTER
- 6.Event.EXIT
- 7.Event.USER\_DEFINED

**set\_name** (*lang=None, string=None*)  
Localized name.

**stage** ()  
Stages changes to GFF structure.

**tag**  
Tag.

**trap\_detect\_dc**  
Trap detect DC.

**trap\_detectable**  
Trap detectable flag.

**trap\_disarm\_dc**  
Trap disarm DC.

**trap\_disarmable**  
Trap disarmable flag.



**trap\_flag**  
Trap flag.

**trap\_one\_shot**  
Trap one shot flag.

**trap\_type**  
Trap type.

**type**  
Type.

**vars**  
Variable table

---

**class** `pynwn.TriggerInstance` (*gff, parent\_obj*)

A trigger instance is one placed in an area in the toolset. As such it's values are derived from its parent GFF structure.

**position**

**stage** ()  
Stages changes to parent GFF structure.

## 1.16 Waypoint

**class** `pynwn.Waypoint` (*resource, instance=False*)

**appearance**  
Appearance ID

**comment**  
Comment.

**get\_description** (*lang=None*)  
Localized description.

**get\_map\_note** (*lang=None*)  
Localized map note.

**get\_name** (*lang=None*)  
Localized name.

**has\_map\_note**  
Has map note flag.

**linked\_to**  
Linked to.

**map\_note\_enabled**  
Map note enabled.

**palette\_id**  
Palette ID.

**resref**  
Resref.

**set\_description** (*lang=None, string=None*)  
Localized description.

**set\_map\_note** (*lang=None, string=None*)  
Localized map note.

**set\_name** (*lang=None, string=None*)  
Localized name.

**stage** ()  
Stage changes to GFF structure.

**tag**  
Tag.

**vars**  
Variable table

---

**class** pynwn.**WaypointInstance** (*gff, parent\_obj*)

**position**

**stage** ()  
Stages changes to parent GFF structure.

---

## Resources

---

### 2.1 Container

**class** `pynwn.Container`

A basic container for ContentObjects

**\_\_getitem\_\_** (*name*)

Get a content object associated with a file name or integer index.

**add** (*content\_obj*)

Add a content object to a container.

**add\_file** (*fname*)

Add a content object from a file to a container.

**get\_content\_data** (*name*)

Get content object data

**Parameters** **name** – File name or int index.

**get\_content\_object** (*name*)

Get content object.

**Parameters** **name** – File name or int index.

**get\_filenames** ()

Gets a list of the filenames of all content objects.

**glob** (*glob\_pattern*)

Returns a list of objects or content objects for file names matching the glob pattern. i.e. Unix shell-style wildcards: \*.utc Note: all file names are converted to lowercase.

**has\_file** (*fname*)

Determines if container has a content object associated with a given filename.

**hashes** (*hash\_type='sha1'*)

Get content object hashes.

**Parameters** **hash\_type** – 'sha1', 'sha256'

**Returns** {filename: hexdigest}

**remove** (*name*)

Remove content object.

**Parameters** **name** – File name

## 2.2 ContentObject

**class** `pynwn.ContentObject` (*resref, res\_type, content=None, offset=None, size=None, abspath=None*)

A ContentObject is an abstraction of any particular NWN resource object either in NWN container (i.e. a hak, mod, or erf) or in a file.

---

**Note:** Parameter `abspath` is ONLY used when the content object is in a DirectoryContainer. Since modifications to content objects are not immediately written to disk, if `io` is changed from a file to `cStringIO`, it's necessary to know where to write the file when `DirectoryContainer.save()` is called.

---

### Parameters

- **resref** – Template resref name.
- **res\_type** – Resource type.
- **content** – Either a file name or `cStringIO`.
- **offset** – Data offset in `io`.
- **size** – Data size.
- **abspath** – Absolute path to the file if one is contained in `io`.

**static from\_file** (*filename*)

Instantiates a ContentObject from a file.

**get** (*mode='rb'*)

Returns the actual data.

**get\_extension** ()

Determines the ContentObject's file extension by resource type.

**get\_filename** ()

Determines the ContentObject's base file name: `<resref>.<ext>`

## 2.3 DirectoryContainer

**class** `pynwn.DirectoryContainer` (*path, only\_nwn=True*)

Bases: `pynwn.resource.Container`

A Container that directly wraps a directory (e.g. override/).

### Parameters

- **path** – Directory path.
- **only\_nwn** – default `True`, If `False` the DirectoryContainer will attempt to load all files, even those that are not NWN resource types.

**\_\_getitem\_\_** (*name*)

Get a content object associated with a file name or integer index.

**add** (*content\_obj*)

Add a content object to a container.

**add\_file** (*fname*)

Add a content object from a file to a container.

**get\_content\_data** (*name*)

Get content object data

**Parameters** **name** – File name or int index.

**get\_content\_object** (*name*)

Get content object.

**Parameters** **name** – File name or int index.

**get\_filenames** ()

Gets a list of the filenames of all content objects.

**glob** (*glob\_pattern*)

Returns a list of objects or content objects for file names matching the glob pattern. i.e. Unix shell-style wildcards: \*.utc Note: all file names are converted to lowercase.

**has\_file** (*fname*)

Determines if container has a content object associated with a given filename.

**hashes** (*hash\_type='sha1'*)

Get content object hashes.

**Parameters** **hash\_type** – 'sha1', 'sha256'

**Returns** {filename: hexdigest}

**remove** (*name*)

Remove content object.

**Parameters** **name** – File name

**save** ()

Saves modified content objects.

---

**Note:** Content objects are not immediately saved to disk when changed. So this function must be called in order to save the them to disk.

---

## 2.4 ResourceManager

**class** pynwn.**ResourceManager**

A container for Container objects.

**\_\_getitem\_\_** (*fname*)

Gets a ContentObject by file name. The order of search is the order in which add\_container was called. I.e. the first added will have the highest priority

**add\_container** (*container*)

Adds a container

**creatures** (*glob=None*)

Returns a list of Creature objects contained in all of the resource managers containers.

**static from\_module** (*mod*, *use\_override=False*, *include\_bioware=True*,  
*path='C:\NeverwinterNights\NWN\')*

Creates a ResourceManager object from a module or module directory.

**Parameters**

- **mod** – Path to module or module directory.

- **use\_override** – default False, If true the overried directory in `path` will be used.
- **include\_bioware** – default True, If false Bioware NWN BIF files will not be used.
- **path** – default “C:NeverwinterNightsNWN”, Path to NWN directory.

---

**Note:**

- If a directory is passed in `mod` it **must** contain a `module.ifo` file.
  - If `include_bioware` is False, `path` can be any working directory that has the same directory stucture as the default NWN installation. I.e. `hak` files are in the subdirectory ‘hak’, overrides in directory ‘override’.
  - When loading the module’s HAKs `.hak` files will attempt to be loaded first. If no file exists, then a directory with the `.hak` files name will attempt to be loaded.
- 

**get\_content\_data** (*fname*)

Gets the contents of a ContentObject that is contained in one of the containers.

**get\_filenames** ()

Gets a list of all file names.

**glob** (*glob\_pattern*)

Returns a list of files matching a glob pattern... i.e. Unix shell-style wildcards: `*.utc` Note: all file names are converted to lowercase.

**has\_file** (*fname*)

Determines if a file exists in one of the containers.

---

## File Formats

---

### 3.1 2da

```
class pynwn.TwoDA(source)
    2da Files.

    __repr__()
        Returns repr of the 2da as a string

    __str__()
        Returns a valid 2da as a string

    get(row, col)
        Gets a 2da entry by row and column label or column index.

    get_column_index(col)
        Gets the column index from a column label.

    get_float(row, col)
        Gets a 2da entry by row and column label or column index as a float.

    get_int(row, col)
        Gets a 2da entry by row and column label or column index as an int.

    parse(content)
        Parses a 2da file.

    set(row, col, val)
        Sets a 2da entry by row and column label or column index. The value passed is automatically coerced to
        str.

    to_ContentType()
        Returns 2da as a ContentType. It's .io contents are cStringIO buffer.

    to_StringIO()
        Returns 2da written in a cStringIO buffer.
```

### 3.2 2dx

```
class pynwn.TwoDX(source)
    2dx Files.
```

**\_\_repr\_\_()**  
Returns repr of the 2dx as a string

**\_\_str\_\_()**  
Returns a valid 2dx as a string

**get(row, col)**  
Gets a 2dx entry by row and column label or column index.

**get\_column\_index(col)**  
Gets the column index from a column label.

**get\_float(row, col)**  
Gets a 2dx entry by row and column label or column index as a float.

**get\_int(row, col)**  
Gets a 2dx entry by row and column label or column index as an int.

**parse(content)**  
Parses a 2dx file.

**set(row, col, val)**  
Sets a 2dx entry by row and column label or column index. The value passed is automatically coerced to str.

**to\_ContentObject()**  
Returns 2dx as a ContentObject. It's .io contents are cStringIO buffer.

**to\_StringIO()**  
Returns 2dx written in a cStringIO buffer.

### 3.3 Erf

**class pynwn.Erf(erf\_type, version='V1.0')**  
Bases: pynwn.resource.Container  
Reads/Writes NWN ERF formats: erf, hak, and mod.

**\_\_getitem\_\_(name)**  
Get a content object associated with a file name or integer index.

**add(content\_obj)**  
Add a content object to a container.

**description(lang=0)**  
Gets description, by language.  
**Parameters lang** – See Bioware's TLK language constants.

**static filename\_length(version)**  
Determine maximum ResRef length.  
**Parameters version** (*str*) – ERF version. Only "V1.0" and "V1.1" are valid parameters.

**static from\_file(fname)**  
Create an Erf from a file handle.  
**Parameters fname** – File name.

**get\_content\_data(name)**  
Get content object data



**Parameters** **name** – File name or int index.

**get\_content\_object** (*name*)

Get content object.

**Parameters** **name** – File name or int index.

**get\_filenames** ()

Gets a list of the filenames of all content objects.

**glob** (*glob\_pattern*)

Returns a list of objects or content objects for file names matching the glob pattern. i.e. Unix shell-style wildcards: \*.utc Note: all file names are converted to lowercase.

**has\_file** (*fname*)

Determines if container has a content object associated with a given filename.

**hashes** (*hash\_type='sha1'*)

Get content object hashes.

**Parameters** **hash\_type** – ‘sha1’, ‘sha256’

**Returns** {filename: hexdigest}

**remove** (*name*)

Remove content object.

**Parameters** **name** – File name

**save** ()

Writes modifications to disk.

If you want to write to some other file use `write_to()`

**set\_description** (*text, lang=0*)

Sets description, by language.

**Parameters**

- **text** – New description.
- **lang** – See Bioware’s TLK language constants.

**write\_to** (*io*)

Writes ERF to file.

**Parameters** **io** – A file path.

## 3.4 Gff

**class** `pynwn.Gff` (*content\_object*)

Represents a GFF file.

**build\_fields** (*structure*)

Build a field list from the specified structure.

**build\_struct** (*sid*)

Builds a structural representation of the specified struct id.

**load** ()

Loads the source of the associated gff file.

**save()**

Saves the current version of the gff structure to the associated file.

**scan**(*offset*, *length*)

Scans the specified length from the specified offset in the open file, then returns to the previous position.

**structure**

Gets the structure, loading it if necessary.

**to\_dict**(*cursor=None*)

Converts gff to dict.

**Returns** A dictionary in the format of nwn-lib json output.

## 3.5 Key / Bif

### 3.5.1 Key

**class** pynwn.**Key**(*fname*, *data\_path*)

...

**Parameters**

- **fname** (*str*) – File handle.
- **data\_path** – Path to your NWN installation directory. e.g: C:/NeverwinterNights/NWN/

### 3.5.2 Bif

The Key class abstracts over Bif classes, so there would never be any reason to instantiate this class directly.

**class** pynwn.**Bif**(*key*, *io*)

Bif.

**has\_res**(*id*)

Determine if Bif contains a resource by an resource ID.

**Parameters** **id** (*int*) – A resource ID.

## 3.6 Tlk

**class** pynwn.**Tlk**(*io=None*)

Loads a TLK file from a file handle.

**\_\_getitem\_\_**(*i*)

Get a TLK element. Tlk supports integer indices and Python slices. Please note that taking a huge slice say a reverse (tlk[::-1]) can be a very costly.

**\_\_len\_\_**()

Determines the highest TLK entry.

**\_\_weakref\_\_**

list of weak references to the object (if defined)

**add**(*text*)

Adds TLK entry to the end of entry list.

**inject** (*other*)

Injects lines from one TLK into another.

## 3.7 Tls

**class** `pynwn.Tls` (*filename=None*)

The Tls file format was created by Meaglyn.

**\_\_weakref\_\_**

list of weak references to the object (if defined)



---

## Checking Local Variables

---

This checks all blueprints and all placed object instances looking for string variables that can be converted to integers or floating point numbers in case of any variable type bugs.

```
#!/usr/bin/env python

from itertools import chain
from pynwn import Module

def check_vars(obj):
    for var, val in obj.vars.string.list():
        try:
            x = int(val)
            print ("    %s: Variable %s (%s) is convertible to int!" % (obj.resref, var, val))
            continue # continue since if it's convertible to int it will also convert to float.
        except:
            pass

        try:
            x = float(val)
            print ("    %s: Variable %s (%s) is convertible to float!" % (obj.resref, var, val))
        except:
            pass

if __name__ == '__main__':
    mod = Module('test.mod')

    print("Checking blueprints...")
    for obj in chain(mod.glob('*.ut[cdeimptw]'), mod.areas):
        check_vars(obj)

    print("\nChecking instances...")
    for area in mod.areas:
        for obj in chain(area.creatures, area.doors, area.placeables,
                        area.triggers, area.stores, area.encounters,
                        area.waypoints, area.items):
            check_vars(obj)
```



---

## Dialog Printer

---

This example prints all the dialogs in a module in tree form.

```
#!/usr/bin/env python

from pynwn import Module

INDENT_WIDTH = 2

def dialog_to_str(dlg):
    def fmt(string, level, link=False):
        justify = INDENT_WIDTH * level
        # Note if the node has no dialog, the string will be None
        if not string:
            res = '<EMPTY>'
        else:
            res = string.strip().replace('\n', '\n'.ljust(justify+1))

        if link:
            res = "Link: " + res

        res = res + '\n'
        return res.rjust(justify + len(res))

    def node_to_str(node, level):
        result = fmt(node.get_text(0), level)
        for ptr in node.pointers:
            node = ptr.get_node(ptr.index)
            # Make sure the dialog pointer isn't a link or else
            # there would be an infinite loop.
            if not ptr.is_link:
                level += 1
                result += node_to_str(node, level)
                level -= 1
            else:
                # Add extra indent for links...
                result += fmt(node.get_text(0), level + 1, True)
        return result

    result = ''
    for start in [s.index for s in dlg.starts]:
        result += node_to_str(dlg.entries[start], 0)

    return result
```

```
if __name__ == '__main__':
    mod = Module('test.mod')

    for dlg in mod.glob('*.dlg'):
        print(dlg.resref)
        print(dialog_to_str(dlg), '\n\n')
```



---

## Minimap Generator

---

Minimap Generator similar to the one in pspeed's nwn java tools.

---

**Note:** This example requires the Python Imaging Library.

---

```
#!/usr/bin/env python

from pynwn import ResourceManager
from pynwn.util.helper import chunks
import ConfigParser, sys
import Image

if __name__ == '__main__':
    mgr = ResourceManager.from_module('test.mod')
    scale = 1
    minimum = 32
    tga_dict = {}

    for area in mgr.module.areas:
        print("Generating minimap for %s" % area.get_name(0))

        config = ConfigParser.ConfigParser()

        try:
            tile_tgas = []
            tga_size = sys.maxint
            config.readfp(mgr[area.tileset + '.set'].to_io())

            for tile in area.tiles:
                tga = config.get('TILE%d' % tile.id, 'ImageMap2D').lower()
                tga_fname = tga+'.tga'
                if not tga_fname in tga_dict:
                    tga_dict[tga_fname] = Image.open(mgr[tga_fname].to_io())

                tga = tga_dict[tga_fname]

                # I chose here to scale all the minimap images to the
                # smallest size so if one is 8x8 they will all be scaled
                # to 8x8.
                tga_size = min(tga_size, tga.size[0])
            tile_tgas.append((tga, tile.orientation))
```

```
# Note: The tile list begins in the bottom left corner
# so I'm going to reverse so that it starts in the top
# left and draw down rather than up.
tile_tgas = chunks(tile_tgas, area.width)[::-1]

# minimum minimap tile size 16x16, just so some of the
# smaller 8x8s are a little larger.
tga_size = max(minimum, tga_size * scale)

new_im = Image.new('RGBA', (area.width * tga_size,
                             area.height * tga_size))

for h in xrange(area.height):
    for w in xrange(area.width):

        im, rot = tile_tgas[h][w]
        new_loc = (w * tga_size, h * tga_size)

        if im.size[0] != tga_size:
            im = im.resize((tga_size, tga_size))

        new_im.paste(im.rotate(rot*90), new_loc)

new_im.save(area.resref + '.png')

except Exception as e:
    print(e)
    continue
```

---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



**p**

pynwn, 3



## Symbols

\_\_getitem\_\_() (pynwn.Container method), 31  
 \_\_getitem\_\_() (pynwn.DirectoryContainer method), 32  
 \_\_getitem\_\_() (pynwn.Erf method), 36  
 \_\_getitem\_\_() (pynwn.ResourceManager method), 33  
 \_\_getitem\_\_() (pynwn.Tlk method), 38  
 \_\_len\_\_() (pynwn.Tlk method), 38  
 \_\_repr\_\_() (pynwn.TwoDA method), 35  
 \_\_repr\_\_() (pynwn.TwoDX method), 35  
 \_\_str\_\_() (pynwn.TwoDA method), 35  
 \_\_str\_\_() (pynwn.TwoDX method), 36  
 \_\_weakref\_\_ (pynwn.Area attribute), 3  
 \_\_weakref\_\_ (pynwn.Tlk attribute), 38  
 \_\_weakref\_\_ (pynwn.Tls attribute), 39

## A

active (pynwn.Encounter attribute), 13  
 active (pynwn.Sound attribute), 24  
 add() (pynwn.Container method), 31  
 add() (pynwn.DirectoryContainer method), 32  
 add() (pynwn.Erf method), 36  
 add() (pynwn.Tlk method), 38  
 add\_container() (pynwn.ResourceManager method), 33  
 add\_file() (pynwn.Container method), 31  
 add\_file() (pynwn.DirectoryContainer method), 32  
 animation\_state (pynwn.Door attribute), 10  
 animation\_state (pynwn.Placeable attribute), 18  
 appearance (pynwn.Creature attribute), 6  
 appearance (pynwn.Door attribute), 10  
 appearance (pynwn.EncounterCreature attribute), 13  
 appearance (pynwn.Placeable attribute), 19  
 appearance (pynwn.PlayerCharacter attribute), 21  
 appearance (pynwn.Waypoint attribute), 29  
 appearance\_head (pynwn.PlayerCharacter attribute), 21  
 Area (class in pynwn), 3  
 areas (pynwn.Module attribute), 17

## B

base\_type (pynwn.Item attribute), 14  
 Bif (class in pynwn), 38

black\_market (pynwn.Store attribute), 26  
 body\_bag (pynwn.Placeable attribute), 19  
 bodybag (pynwn.Creature attribute), 6  
 bodybag (pynwn.PlayerCharacter attribute), 21  
 build\_fields() (pynwn.Gff method), 37  
 build\_struct() (pynwn.Gff method), 37

## C

chance (pynwn.ItemProperty attribute), 16  
 charges (pynwn.Item attribute), 14  
 charisma (pynwn.Creature attribute), 6  
 charisma (pynwn.PlayerCharacter attribute), 21  
 classes (pynwn.Creature attribute), 6  
 comment (pynwn.Area attribute), 3  
 comment (pynwn.Creature attribute), 6  
 comment (pynwn.Door attribute), 10  
 comment (pynwn.Encounter attribute), 13  
 comment (pynwn.Item attribute), 15  
 comment (pynwn.Placeable attribute), 19  
 comment (pynwn.Sound attribute), 24  
 comment (pynwn.Store attribute), 26  
 comment (pynwn.Trigger attribute), 27  
 comment (pynwn.Waypoint attribute), 29  
 constitution (pynwn.Creature attribute), 6  
 constitution (pynwn.PlayerCharacter attribute), 22  
 Container (class in pynwn), 31  
 ContentObject (class in pynwn), 32  
 continous (pynwn.Sound attribute), 24  
 conversation (pynwn.Creature attribute), 6  
 conversation (pynwn.Door attribute), 10  
 conversation (pynwn.Placeable attribute), 19  
 conversation (pynwn.PlayerCharacter attribute), 22  
 corpse\_decay (pynwn.Creature attribute), 6  
 corpse\_decay (pynwn.PlayerCharacter attribute), 22  
 cost (pynwn.Item attribute), 15  
 cost\_additional (pynwn.Item attribute), 15  
 cost\_table (pynwn.ItemProperty attribute), 16  
 cost\_value (pynwn.ItemProperty attribute), 16  
 cr (pynwn.Creature attribute), 6  
 cr (pynwn.EncounterCreature attribute), 13  
 cr (pynwn.PaletteNode attribute), 18

cr (pynwn.PlayerCharacter attribute), 22  
 Creature (class in pynwn), 6  
 CreatureInstance (class in pynwn), 9  
 creatures (pynwn.Area attribute), 3  
 creatures (pynwn.Encounter attribute), 13  
 creatures() (pynwn.ResourceManager method), 33  
 cursed (pynwn.Item attribute), 15

## D

day\_ambient\_color (pynwn.Area attribute), 3  
 day\_diffuse\_color (pynwn.Area attribute), 3  
 day\_fog\_amount (pynwn.Area attribute), 3  
 day\_fog\_color (pynwn.Area attribute), 3  
 day\_night\_cycle (pynwn.Area attribute), 3  
 day\_shadows (pynwn.Area attribute), 3  
 dead\_selectable (pynwn.PlayerCharacter attribute), 22  
 delay\_entry (pynwn.Dialog attribute), 9  
 desc\_override (pynwn.PlayerCharacter attribute), 22  
 description() (pynwn.Erf method), 36  
 detect\_mode (pynwn.PlayerCharacter attribute), 22  
 dexterity (pynwn.Creature attribute), 6  
 dexterity (pynwn.PlayerCharacter attribute), 22  
 Dialog (class in pynwn), 9  
 diety (pynwn.Creature attribute), 6  
 diety (pynwn.PlayerCharacter attribute), 22  
 difficulty (pynwn.Encounter attribute), 13  
 difficulty\_index (pynwn.Encounter attribute), 13  
 DirectoryContainer (class in pynwn), 32  
 disarmable (pynwn.Creature attribute), 6  
 disarmable (pynwn.PlayerCharacter attribute), 22  
 display\_name (pynwn.Item attribute), 15  
 distance\_max (pynwn.Sound attribute), 25  
 distance\_min (pynwn.Sound attribute), 25  
 Door (class in pynwn), 10  
 DoorInstance (class in pynwn), 12  
 doors (pynwn.Area attribute), 3  
 dropable (pynwn.RepositoryItem attribute), 16

## E

elevation (pynwn.Sound attribute), 25  
 Encounter (class in pynwn), 13  
 EncounterCreature (class in pynwn), 13  
 EncounterInstance (class in pynwn), 14  
 encounters (pynwn.Area attribute), 3  
 entries (pynwn.Dialog attribute), 9  
 entry\_area (pynwn.Module attribute), 17  
 entry\_location (pynwn.Module attribute), 17  
 equips (pynwn.Creature attribute), 6  
 equips (pynwn.CreatureInstance attribute), 9  
 equips (pynwn.PlayerCharacter attribute), 22  
 Erf (class in pynwn), 36  
 expansion\_pack (pynwn.Module attribute), 17  
 experience (pynwn.PlayerCharacter attribute), 22

## F

Faction (class in pynwn), 14  
 faction (pynwn.Creature attribute), 6  
 faction (pynwn.Door attribute), 10  
 faction (pynwn.Encounter attribute), 13  
 faction (pynwn.PaletteNode attribute), 18  
 faction (pynwn.Placeable attribute), 19  
 faction (pynwn.PlayerCharacter attribute), 22  
 faction (pynwn.Trigger attribute), 27  
 factions (pynwn.Faction attribute), 14  
 feats (pynwn.Creature attribute), 6  
 filename\_length() (pynwn.Erf static method), 36  
 flags (pynwn.Area attribute), 3  
 fog\_clip\_distance (pynwn.Area attribute), 3  
 from\_file() (pynwn.ContentObject static method), 32  
 from\_file() (pynwn.Erf static method), 36  
 from\_module() (pynwn.ResourceManager static method), 33

## G

game\_version (pynwn.Module attribute), 17  
 gender (pynwn.Creature attribute), 6  
 gender (pynwn.PlayerCharacter attribute), 22  
 generic\_type (pynwn.Door attribute), 10  
 get() (pynwn.ContentObject method), 32  
 get() (pynwn.TwoDA method), 35  
 get() (pynwn.TwoDX method), 36  
 get\_column\_index() (pynwn.TwoDA method), 35  
 get\_column\_index() (pynwn.TwoDX method), 36  
 get\_content\_data() (pynwn.Container method), 31  
 get\_content\_data() (pynwn.DirectoryContainer method), 32  
 get\_content\_data() (pynwn.Erf method), 36  
 get\_content\_data() (pynwn.ResourceManager method), 34  
 get\_content\_object() (pynwn.Container method), 31  
 get\_content\_object() (pynwn.DirectoryContainer method), 33  
 get\_content\_object() (pynwn.Erf method), 37  
 get\_description() (pynwn.Creature method), 7  
 get\_description() (pynwn.Door method), 10  
 get\_description() (pynwn.Item method), 15  
 get\_description() (pynwn.Module method), 17  
 get\_description() (pynwn.Placeable method), 19  
 get\_description() (pynwn.PlayerCharacter method), 22  
 get\_description() (pynwn.Store method), 26  
 get\_description() (pynwn.Waypoint method), 29  
 get\_description\_id() (pynwn.Item method), 15  
 get\_description\_id() (pynwn.Store method), 26  
 get\_extension() (pynwn.ContentObject method), 32  
 get\_filename() (pynwn.ContentObject method), 32  
 get\_filenames() (pynwn.Container method), 31  
 get\_filenames() (pynwn.DirectoryContainer method), 33  
 get\_filenames() (pynwn.Erf method), 37



get\_filenames() (pynwn.ResourceManager method), 34  
 get\_float() (pynwn.TwoDA method), 35  
 get\_float() (pynwn.TwoDX method), 36  
 get\_int() (pynwn.TwoDA method), 35  
 get\_int() (pynwn.TwoDX method), 36  
 get\_map\_note() (pynwn.Waypoint method), 29  
 get\_name() (pynwn.Area method), 4  
 get\_name() (pynwn.Door method), 10  
 get\_name() (pynwn.Encounter method), 13  
 get\_name() (pynwn.Item method), 15  
 get\_name() (pynwn.Module method), 17  
 get\_name() (pynwn.Placeable method), 19  
 get\_name() (pynwn.Sound method), 25  
 get\_name() (pynwn.Store method), 26  
 get\_name() (pynwn.Trigger method), 27  
 get\_name() (pynwn.Waypoint method), 29  
 get\_name\_first() (pynwn.Creature method), 7  
 get\_name\_first() (pynwn.PlayerCharacter method), 22  
 get\_name\_last() (pynwn.Creature method), 7  
 get\_name\_last() (pynwn.PlayerCharacter method), 22  
 get\_skill() (pynwn.Creature method), 7  
 Gff (class in pynwn), 37  
 glob() (pynwn.Container method), 31  
 glob() (pynwn.DirectoryContainer method), 33  
 glob() (pynwn.Erf method), 37  
 glob() (pynwn.Module method), 17  
 glob() (pynwn.ResourceManager method), 34  
 gold (pynwn.Store attribute), 26  
 goodevil (pynwn.Creature attribute), 7  
 goodevil (pynwn.PlayerCharacter attribute), 22

## H

haks (pynwn.Module attribute), 17  
 hardness (pynwn.Door attribute), 10  
 hardness (pynwn.Placeable attribute), 19  
 has\_file() (pynwn.Container method), 31  
 has\_file() (pynwn.DirectoryContainer method), 33  
 has\_file() (pynwn.Erf method), 37  
 has\_file() (pynwn.ResourceManager method), 34  
 has\_inventory (pynwn.Placeable attribute), 19  
 has\_map\_note (pynwn.Waypoint attribute), 29  
 has\_res() (pynwn.Bif method), 38  
 hashes() (pynwn.Container method), 31  
 hashes() (pynwn.DirectoryContainer method), 33  
 hashes() (pynwn.Erf method), 37  
 height (pynwn.Area attribute), 4  
 highlight\_height (pynwn.Trigger attribute), 27  
 hours (pynwn.Sound attribute), 25  
 hp (pynwn.Creature attribute), 7  
 hp (pynwn.Door attribute), 10  
 hp (pynwn.Placeable attribute), 19  
 hp (pynwn.PlayerCharacter attribute), 22  
 hp\_current (pynwn.Creature attribute), 7  
 hp\_current (pynwn.Door attribute), 11

hp\_current (pynwn.Placeable attribute), 19  
 hp\_current (pynwn.PlayerCharacter attribute), 22  
 hp\_max (pynwn.Creature attribute), 7  
 hp\_max (pynwn.PlayerCharacter attribute), 22  
 hp\_pregame\_current (pynwn.PlayerCharacter attribute), 23

## I

id (pynwn.PaletteNode attribute), 18  
 identified (pynwn.Item attribute), 15  
 infinite (pynwn.RepositoryItem attribute), 16  
 inject() (pynwn.Tlk method), 38  
 intelligence (pynwn.Creature attribute), 7  
 intelligence (pynwn.PlayerCharacter attribute), 23  
 interruptable (pynwn.Creature attribute), 7  
 interruptable (pynwn.Door attribute), 11  
 interruptable (pynwn.Placeable attribute), 19  
 interval (pynwn.Sound attribute), 25  
 interval\_variation (pynwn.Sound attribute), 25  
 is\_commandable (pynwn.PlayerCharacter attribute), 23  
 is\_destroyable (pynwn.PlayerCharacter attribute), 23  
 is\_dm (pynwn.PlayerCharacter attribute), 23  
 is\_immortal (pynwn.Creature attribute), 7  
 is\_immortal (pynwn.PlayerCharacter attribute), 23  
 is\_leaf (pynwn.PaletteNode attribute), 18  
 is\_night (pynwn.Area attribute), 4  
 is\_pc (pynwn.Creature attribute), 7  
 is\_pc (pynwn.PlayerCharacter attribute), 23  
 is\_raisable (pynwn.PlayerCharacter attribute), 23  
 Item (class in pynwn), 14  
 ItemInstance (class in pynwn), 16  
 ItemProperty (class in pynwn), 16  
 items (pynwn.Area attribute), 4  
 items (pynwn.Creature attribute), 7  
 items (pynwn.CreatureInstance attribute), 9  
 items (pynwn.Placeable attribute), 19  
 items (pynwn.PlaceableInstance attribute), 21  
 items (pynwn.Store attribute), 26  
 items (pynwn.StoreInstance attribute), 27

## J

Journal (class in pynwn), 16

## K

Key (class in pynwn), 38  
 key\_auto\_remove (pynwn.Door attribute), 11  
 key\_auto\_remove (pynwn.Placeable attribute), 19  
 key\_auto\_remove (pynwn.Trigger attribute), 28  
 key\_required (pynwn.Door attribute), 11  
 key\_required (pynwn.Placeable attribute), 19  
 key\_tag (pynwn.Door attribute), 11  
 key\_tag (pynwn.Placeable attribute), 19  
 key\_tag (pynwn.Trigger attribute), 28

## L

lawchaos (pynwn.Creature attribute), 7  
lawchaos (pynwn.PlayerCharacter attribute), 23  
level\_stats (pynwn.PlayerCharacter attribute), 23  
lighting\_scheme (pynwn.Area attribute), 4  
lightning (pynwn.Area attribute), 4  
linked\_to (pynwn.Door attribute), 11  
linked\_to (pynwn.Trigger attribute), 28  
linked\_to (pynwn.Waypoint attribute), 29  
linked\_to\_flags (pynwn.Door attribute), 11  
linked\_to\_flags (pynwn.Trigger attribute), 28  
listen\_modifier (pynwn.Area attribute), 4  
load() (pynwn.Gff method), 37  
load\_screen (pynwn.Area attribute), 4  
load\_screen (pynwn.Door attribute), 11  
load\_screen (pynwn.Trigger attribute), 28  
lock\_close\_dc (pynwn.Door attribute), 11  
lock\_close\_dc (pynwn.Placeable attribute), 19  
lock\_open\_dc (pynwn.Door attribute), 11  
lock\_open\_dc (pynwn.Placeable attribute), 19  
lockable (pynwn.Door attribute), 11  
lockable (pynwn.Placeable attribute), 19  
locked (pynwn.Door attribute), 11  
locked (pynwn.Placeable attribute), 19  
looping (pynwn.Sound attribute), 25  
lootable (pynwn.Creature attribute), 7  
lootable (pynwn.PlayerCharacter attribute), 23

## M

map\_note\_enabled (pynwn.Waypoint attribute), 29  
mark\_down (pynwn.Store attribute), 26  
mark\_down\_bm (pynwn.Store attribute), 26  
mark\_up (pynwn.Store attribute), 26  
master (pynwn.PlayerCharacter attribute), 23  
max\_creatures (pynwn.Encounter attribute), 13  
Module (class in pynwn), 17  
movement\_rate (pynwn.PlayerCharacter attribute), 23

## N

name (pynwn.PaletteNode attribute), 18  
natural\_ac (pynwn.Creature attribute), 7  
natural\_ac (pynwn.PlayerCharacter attribute), 23  
night\_ambient\_color (pynwn.Area attribute), 4  
night\_diffuse\_color (pynwn.Area attribute), 4  
night\_fog\_amount (pynwn.Area attribute), 4  
night\_fog\_color (pynwn.Area attribute), 4  
night\_shadows (pynwn.Area attribute), 4  
no\_perm\_death (pynwn.Creature attribute), 7  
no\_perm\_death (pynwn.PlayerCharacter attribute), 23  
no\_rest (pynwn.Area attribute), 4  
nodes (pynwn.Palette attribute), 18  
nodes (pynwn.PaletteNode attribute), 18

## P

Palette (class in pynwn), 18  
palette\_id (pynwn.Creature attribute), 7  
palette\_id (pynwn.Encounter attribute), 13  
palette\_id (pynwn.Item attribute), 15  
palette\_id (pynwn.Sound attribute), 25  
palette\_id (pynwn.Store attribute), 26  
palette\_id (pynwn.Waypoint attribute), 29  
paletted\_id (pynwn.Door attribute), 11  
paletted\_id (pynwn.Placeable attribute), 19  
paletted\_id (pynwn.Trigger attribute), 28  
PaletteNode (class in pynwn), 18  
param\_table (pynwn.ItemProperty attribute), 16  
param\_value (pynwn.ItemProperty attribute), 16  
parse() (pynwn.TwoDA method), 35  
parse() (pynwn.TwoDX method), 36  
perception\_range (pynwn.Creature attribute), 7  
phenotype (pynwn.Creature attribute), 7  
phenotype (pynwn.PlayerCharacter attribute), 23  
pitch\_variation (pynwn.Sound attribute), 25  
Placeable (class in pynwn), 18  
PlaceableInstance (class in pynwn), 21  
placeables (pynwn.Area attribute), 4  
player\_only (pynwn.Encounter attribute), 13  
PlayerCharacter (class in pynwn), 21  
plot (pynwn.Door attribute), 11  
plot (pynwn.Item attribute), 15  
plot (pynwn.Placeable attribute), 20  
plot (pynwn.PlayerCharacter attribute), 23  
portrait (pynwn.PlayerCharacter attribute), 23  
portrait\_id (pynwn.Creature attribute), 7  
portrait\_id (pynwn.Door attribute), 11  
portrait\_id (pynwn.Placeable attribute), 20  
portrait\_id (pynwn.PlayerCharacter attribute), 23  
portrait\_id (pynwn.Trigger attribute), 28  
position (pynwn.DoorInstance attribute), 12  
position (pynwn.PlaceableInstance attribute), 21  
position (pynwn.RepositoryItem attribute), 16  
position (pynwn.TriggerInstance attribute), 29  
position (pynwn.WaypointInstance attribute), 30  
positional (pynwn.Sound attribute), 25  
prevent\_zoom (pynwn.Dialog attribute), 9  
price\_id (pynwn.Store attribute), 26  
price\_max\_buy (pynwn.Store attribute), 26  
priority (pynwn.Sound attribute), 25  
properties (pynwn.Item attribute), 15  
pvp (pynwn.Area attribute), 4  
pynwn (module), 3

## Q

quests (pynwn.Journal attribute), 16

## R

race (pynwn.Creature attribute), 8

race (pynwn.PlayerCharacter attribute), 23  
 rain (pynwn.Area attribute), 4  
 random (pynwn.Sound attribute), 25  
 random\_position (pynwn.Sound attribute), 25  
 random\_range (pynwn.Sound attribute), 25  
 rec\_creatures (pynwn.Encounter attribute), 13  
 remove() (pynwn.Container method), 31  
 remove() (pynwn.DirectoryContainer method), 33  
 remove() (pynwn.Erf method), 37  
 replies (pynwn.Dialog attribute), 9  
 RepositoryItem (class in pynwn), 16  
 reputations (pynwn.Faction attribute), 14  
 reset (pynwn.Encounter attribute), 13  
 reset\_time (pynwn.Encounter attribute), 13  
 ResourceManager (class in pynwn), 33  
 respawns (pynwn.Encounter attribute), 14  
 resref (pynwn.Area attribute), 4  
 resref (pynwn.Creature attribute), 8  
 resref (pynwn.Dialog attribute), 10  
 resref (pynwn.Door attribute), 11  
 resref (pynwn.Encounter attribute), 14  
 resref (pynwn.EncounterCreature attribute), 13  
 resref (pynwn.Item attribute), 15  
 resref (pynwn.PaletteNode attribute), 18  
 resref (pynwn.Placeable attribute), 20  
 resref (pynwn.RepositoryItem attribute), 16  
 resref (pynwn.Sound attribute), 25  
 resref (pynwn.Store attribute), 26  
 resref (pynwn.Trigger attribute), 28  
 resref (pynwn.Waypoint attribute), 29

## S

save() (pynwn.DirectoryContainer method), 33  
 save() (pynwn.Erf method), 37  
 save() (pynwn.Gff method), 37  
 save\_fortitude (pynwn.Creature attribute), 8  
 save\_fortitude (pynwn.Door attribute), 11  
 save\_fortitude (pynwn.Placeable attribute), 20  
 save\_fortitude (pynwn.PlayerCharacter attribute), 23  
 save\_reflex (pynwn.Creature attribute), 8  
 save\_reflex (pynwn.Door attribute), 11  
 save\_reflex (pynwn.Placeable attribute), 20  
 save\_reflex (pynwn.PlayerCharacter attribute), 24  
 save\_will (pynwn.Creature attribute), 8  
 save\_will (pynwn.Door attribute), 11  
 save\_will (pynwn.Placeable attribute), 20  
 save\_will (pynwn.PlayerCharacter attribute), 24  
 scan() (pynwn.Gff method), 38  
 script\_abort (pynwn.Dialog attribute), 10  
 script\_end (pynwn.Dialog attribute), 10  
 scripts (pynwn.Area attribute), 4  
 scripts (pynwn.Creature attribute), 8  
 scripts (pynwn.Door attribute), 11  
 scripts (pynwn.Encounter attribute), 14  
 scripts (pynwn.Module attribute), 17  
 scripts (pynwn.Placeable attribute), 20  
 scripts (pynwn.Store attribute), 26  
 scripts (pynwn.Trigger attribute), 28  
 set() (pynwn.TwoDA method), 35  
 set() (pynwn.TwoDX method), 36  
 set\_description() (pynwn.Creature method), 8  
 set\_description() (pynwn.Door method), 12  
 set\_description() (pynwn.Erf method), 37  
 set\_description() (pynwn.Item method), 15  
 set\_description() (pynwn.Module method), 18  
 set\_description() (pynwn.Placeable method), 20  
 set\_description() (pynwn.PlayerCharacter method), 24  
 set\_description() (pynwn.Store method), 27  
 set\_description() (pynwn.Waypoint method), 29  
 set\_description\_id() (pynwn.Item method), 15  
 set\_description\_id() (pynwn.Store method), 27  
 set\_map\_note() (pynwn.Waypoint method), 30  
 set\_name() (pynwn.Area method), 5  
 set\_name() (pynwn.Door method), 12  
 set\_name() (pynwn.Encounter method), 14  
 set\_name() (pynwn.Item method), 15  
 set\_name() (pynwn.Module method), 18  
 set\_name() (pynwn.Placeable method), 20  
 set\_name() (pynwn.Sound method), 25  
 set\_name() (pynwn.Store method), 27  
 set\_name() (pynwn.Trigger method), 28  
 set\_name() (pynwn.Waypoint method), 30  
 set\_name\_first() (pynwn.Creature method), 8  
 set\_name\_first() (pynwn.PlayerCharacter method), 24  
 set\_name\_last() (pynwn.Creature method), 8  
 set\_name\_last() (pynwn.PlayerCharacter method), 24  
 set\_skill() (pynwn.Creature method), 8  
 shadow\_opacity (pynwn.Area attribute), 5  
 size (pynwn.PlayerCharacter attribute), 24  
 skillpoints (pynwn.PlayerCharacter attribute), 24  
 skills (pynwn.Creature attribute), 8  
 skybox (pynwn.Area attribute), 5  
 snow (pynwn.Area attribute), 5  
 Sound (class in pynwn), 24  
 SoundInstance (class in pynwn), 26  
 sounds (pynwn.Area attribute), 5  
 sounds (pynwn.Sound attribute), 25  
 spawn\_option (pynwn.Encounter attribute), 14  
 special\_abilities (pynwn.Creature attribute), 8  
 spot\_modifier (pynwn.Area attribute), 5  
 stack\_size (pynwn.Item attribute), 15  
 stage() (pynwn.Creature method), 8  
 stage() (pynwn.CreatureInstance method), 9  
 stage() (pynwn.Dialog method), 10  
 stage() (pynwn.Faction method), 14  
 stage() (pynwn.Module method), 18  
 stage() (pynwn.Palette method), 18  
 stage() (pynwn.PaletteNode method), 18

stage() (pynwn.Placeable method), 20  
 stage() (pynwn.PlaceableInstance method), 21  
 stage() (pynwn.PlayerCharacter method), 24  
 stage() (pynwn.Sound method), 25  
 stage() (pynwn.SoundInstance method), 26  
 stage() (pynwn.Store method), 27  
 stage() (pynwn.StoreInstance method), 27  
 stage() (pynwn.Trigger method), 28  
 stage() (pynwn.TriggerInstance method), 29  
 stage() (pynwn.Waypoint method), 30  
 stage() (pynwn.WaypointInstance method), 30  
 starting\_package (pynwn.Creature attribute), 8  
 starting\_package (pynwn.PlayerCharacter attribute), 24  
 starts (pynwn.Dialog attribute), 10  
 static (pynwn.Placeable attribute), 20  
 stealth\_mode (pynwn.PlayerCharacter attribute), 24  
 stolen (pynwn.Item attribute), 15  
 Store (class in pynwn), 26  
 StoreInstance (class in pynwn), 27  
 stores (pynwn.Area attribute), 5  
 strength (pynwn.Creature attribute), 9  
 strength (pynwn.PlayerCharacter attribute), 24  
 strref (pynwn.PaletteNode attribute), 18  
 structure (pynwn.Gff attribute), 38  
 subrace (pynwn.Creature attribute), 9  
 subrace (pynwn.PlayerCharacter attribute), 24  
 subtype (pynwn.ItemProperty attribute), 16

## T

tag (pynwn.Area attribute), 5  
 tag (pynwn.Creature attribute), 9  
 tag (pynwn.Door attribute), 12  
 tag (pynwn.Encounter attribute), 14  
 tag (pynwn.Item attribute), 15  
 tag (pynwn.Placeable attribute), 20  
 tag (pynwn.PlayerCharacter attribute), 24  
 tag (pynwn.Sound attribute), 25  
 tag (pynwn.Store attribute), 27  
 tag (pynwn.Trigger attribute), 28  
 tag (pynwn.Waypoint attribute), 30  
 tail (pynwn.Creature attribute), 9  
 tail (pynwn.PlayerCharacter attribute), 24  
 tiles (pynwn.Area attribute), 5  
 tileset (pynwn.Area attribute), 5  
 times (pynwn.Sound attribute), 25  
 Tlk (class in pynwn), 38  
 tlk (pynwn.Module attribute), 18  
 Tls (class in pynwn), 39  
 to\_ContentObject() (pynwn.TwoDA method), 35  
 to\_ContentObject() (pynwn.TwoDX method), 36  
 to\_dict() (pynwn.Gff method), 38  
 to\_StringIO() (pynwn.TwoDA method), 35  
 to\_StringIO() (pynwn.TwoDX method), 36  
 trap\_detect\_dc (pynwn.Door attribute), 12

trap\_detect\_dc (pynwn.Placeable attribute), 21  
 trap\_detect\_dc (pynwn.Trigger attribute), 28  
 trap\_detectable (pynwn.Door attribute), 12  
 trap\_detectable (pynwn.Placeable attribute), 21  
 trap\_detectable (pynwn.Trigger attribute), 28  
 trap\_disarm\_dc (pynwn.Door attribute), 12  
 trap\_disarm\_dc (pynwn.Placeable attribute), 21  
 trap\_disarm\_dc (pynwn.Trigger attribute), 28  
 trap\_disarmable (pynwn.Door attribute), 12  
 trap\_disarmable (pynwn.Placeable attribute), 21  
 trap\_disarmable (pynwn.Trigger attribute), 28  
 trap\_flag (pynwn.Door attribute), 12  
 trap\_flag (pynwn.Placeable attribute), 21  
 trap\_flag (pynwn.Trigger attribute), 28  
 trap\_one\_shot (pynwn.Door attribute), 12  
 trap\_one\_shot (pynwn.Placeable attribute), 21  
 trap\_one\_shot (pynwn.Trigger attribute), 29  
 trap\_type (pynwn.Door attribute), 12  
 trap\_type (pynwn.Placeable attribute), 21  
 trap\_type (pynwn.Trigger attribute), 29  
 Trigger (class in pynwn), 27  
 TriggerInstance (class in pynwn), 29  
 triggers (pynwn.Area attribute), 5  
 TwoDA (class in pynwn), 35  
 TwoDX (class in pynwn), 35  
 type (pynwn.ItemProperty attribute), 16  
 type (pynwn.Placeable attribute), 21  
 type (pynwn.Trigger attribute), 29

## U

unique (pynwn.EncounterCreature attribute), 13  
 useable (pynwn.Placeable attribute), 21

## V

vars (pynwn.Area attribute), 5  
 vars (pynwn.Creature attribute), 9  
 vars (pynwn.Door attribute), 12  
 vars (pynwn.Encounter attribute), 14  
 vars (pynwn.Item attribute), 15  
 vars (pynwn.Module attribute), 18  
 vars (pynwn.Placeable attribute), 21  
 vars (pynwn.Store attribute), 27  
 vars (pynwn.Trigger attribute), 29  
 vars (pynwn.Waypoint attribute), 30  
 version (pynwn.Area attribute), 5  
 volume (pynwn.Sound attribute), 25  
 volume\_variation (pynwn.Sound attribute), 25

## W

walkrate (pynwn.Creature attribute), 9  
 Waypoint (class in pynwn), 29  
 WaypointInstance (class in pynwn), 30  
 waypoints (pynwn.Area attribute), 5  
 width (pynwn.Area attribute), 5

[will\\_not\\_buy \(pynwn.Store attribute\), 27](#)  
[will\\_only\\_buy \(pynwn.Store attribute\), 27](#)  
[wind\\_power \(pynwn.Area attribute\), 5](#)  
[wings \(pynwn.Creature attribute\), 9](#)  
[wings \(pynwn.PlayerCharacter attribute\), 24](#)  
[wisdom \(pynwn.Creature attribute\), 9](#)  
[wisdom \(pynwn.PlayerCharacter attribute\), 24](#)  
[word\\_count \(pynwn.Dialog attribute\), 10](#)  
[write\\_to\(\) \(pynwn.Erf method\), 37](#)

## X

[xp\\_scale \(pynwn.Module attribute\), 18](#)